PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 9/44 — A2

(11) International Publication Number: **WO 99/61983**

(43) International Publication Date: 2 December 1999 (02.12.99)

(21) International Application Number: PCT/SE99/00892

(22) International Filing Date: 26 May 1999 (26.05.99)

(30) Priority Data:
9801897–1 28 May 1998 (28.05.98) SE

(71) Applicant *(for all designated States except US)*: TELEFONAKTIEBOLAGET LM ERICSSON (publ) [SE/SE]; S–126 25 Stockholm (SE).

(72) Inventors; and
(75) Inventors/Applicants *(for US only)*: AQUILON, Adam [SE/SE]; Margaretavägen 18, S–177 56 Järfälla (SE). WILLEHADSON, Stefan [SE/SE]; Spetsvägen 22, S–757 57 Uppsala (SE). HALL, Hans [SE/SE]; Studentvägen 22:46, S–752 34 Uppsala (SE).

(74) Agents: BERG, S., A. et al.; Albihns Patentbyrå Stockholm AB, P.O. Box 5581, S–114 85 Stockholm (SE).

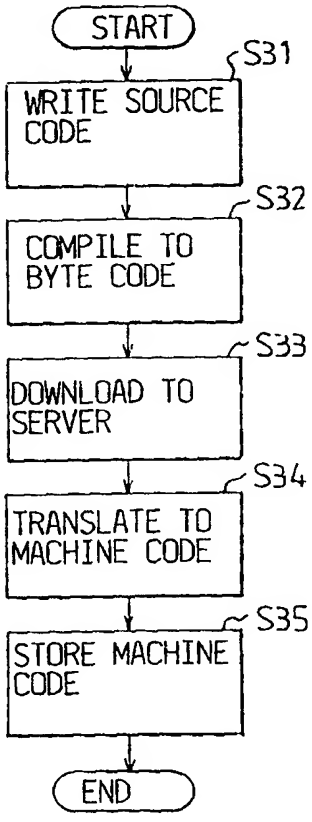
(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, US, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: CHANGING FUNCTIONALITY OF A MODULE TERMINAL IN A WIRELESS NETWORK

(57) Abstract

The functionality of portable wireless devices can be changed by providing programs written in a language, e.g. Java, producing a verifiable intermediary code (byte code). The intermediary code is translated to machine code in a node in the network before it is transferred to the wireless device on which it is to be executed. Several machine code variants of each program, translated using translators for different types of terminals, may exist. This enables use of the same source code for different terminal types and a division between software vendor, terminal vendor and network operator of the responsibility for carrying out the different functions involved in changing the functionality of a device, namely developing the source code of the program, compiling the source code into byte code, translating the byte code into machine code and transferring the machine code to the terminals on which it is to be run.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	**ES**	Spain	**LS**	Lesotho	**SI**	Slovenia
AM	Armenia	**FI**	Finland	**LT**	Lithuania	**SK**	Slovakia
AT	Austria	**FR**	France	**LU**	Luxembourg	**SN**	Senegal
AU	Australia	**GA**	Gabon	**LV**	Latvia	**SZ**	Swaziland
AZ	Azerbaijan	**GB**	United Kingdom	**MC**	Monaco	**TD**	Chad
BA	Bosnia and Herzegovina	**GE**	Georgia	**MD**	Republic of Moldova	**TG**	Togo
BB	Barbados	**GH**	Ghana	**MG**	Madagascar	**TJ**	Tajikistan
BE	Belgium	**GN**	Guinea	**MK**	The former Yugoslav Republic of Macedonia	**TM**	Turkmenistan
BF	Burkina Faso	**GR**	Greece			**TR**	Turkey
BG	Bulgaria	**HU**	Hungary	**ML**	Mali	**TT**	Trinidad and Tobago
BJ	Benin	**IE**	Ireland	**MN**	Mongolia	**UA**	Ukraine
BR	Brazil	**IL**	Israel	**MR**	Mauritania	**UG**	Uganda
BY	Belarus	**IS**	Iceland	**MW**	Malawi	**US**	United States of America
CA	Canada	**IT**	Italy	**MX**	Mexico	**UZ**	Uzbekistan
CF	Central African Republic	**JP**	Japan	**NE**	Niger	**VN**	Viet Nam
CG	Congo	**KE**	Kenya	**NL**	Netherlands	**YU**	Yugoslavia
CH	Switzerland	**KG**	Kyrgyzstan	**NO**	Norway	**ZW**	Zimbabwe
CI	Côte d'Ivoire	**KP**	Democratic People's Republic of Korea	**NZ**	New Zealand		
CM	Cameroon			**PL**	Poland		
CN	China	**KR**	Republic of Korea	**PT**	Portugal		
CU	Cuba	**KZ**	Kazakstan	**RO**	Romania		
CZ	Czech Republic	**LC**	Saint Lucia	**RU**	Russian Federation		
DE	Germany	**LI**	Liechtenstein	**SD**	Sudan		
DK	Denmark	**LK**	Sri Lanka	**SE**	Sweden		
EE	Estonia	**LR**	Liberia	**SG**	Singapore		

CHANGING FUNCTIONALITY OF A MODULE TERMINAL IN A WIRELESS NETWORK

Technical Field

The present invention relates to service provisioning, and in particular to service provisioning to handheld or portable terminals in wireless network.

Description of Related Art

The use of mobile telephones for different types of purposes is increasing. New types of mobile or portable terminals have been introduced, such as smart phones, message pads, personal digital assistants, intelligent pagers, etc. New terminals with increased functionality are constantly being developed. The possibility of upgrading the functionality of an existing portable terminal is, however, limited with current technology.

Thus, there is a desire to be able to add functions to existing portable terminals and to tailor the functionality to suit the user's needs. There is a general requirement for portable terminals to be as small and light as possible. This limits the memory space available, as well as the processing speed and the power. Other factors, for example, the limited battery capacity, and the need for cooling must also be taken into account.

Current development of portable devices tends to place system intelligence in the portable device itself, rather than in a server, in order to make user interaction faster and to reduce the bandwidth required to transfer information and functionality. There is, however, a limit to the amount of memory and processor capacity that can be included in a portable device without increasing its size and weight too much. Thus, when designing a mobile terminal, a balance must be found between functionality requirement and the size and weight, etc., of the terminal. When determining what to store in the mobile terminal and what to store in a node in the network, a balance must be found between the processor capacity and memory space needed in

the portable terminal and the bandwidth required to transfer programs and/or information between the network and the portable terminal.

Also, a large number of different terminal types exist, using different hardware and software platforms. Therefore, it is desirable to make programs that are executable on several different platforms.

Summary of the Invention

It is an object of the invention to enable the execution of more advanced, or demanding, programs in a portable device than would normally be possible with respect to the processor capacity and memory space of the portable device.

It is an object of the invention to enable the execution of programs in a small device with limited processor capacity and/or memory space without reducing the processing speed.

It is another object of the present invention to enable the addition of new functions, upgrading of functions or the complete replacement of a function package in different types of handheld devices, especially handheld devices adapted for use in a cellular network.

It is yet another object of the present invention to enable the execution of programs that do not have to be permanently stored in the terminal.

These objects are achieved according to the present invention, by a method of changing the functionality of a mobile terminal connected by a wireless connection in a network,

said functionality being implemented as programs in a programming language for which the source code may be compiled to an intermediary code, said intermediary code being executable and verifiable,

comprising the following steps:

- translating said program into machine code using at least one translator in dependence of the hardware and software environment of at least one terminal type;
- downloading the translated program to said mobile terminal.

The objects are also achieved by an apparatus for providing programs in a network comprising hosts and several different types of terminals, at least some of said terminals being connected by wireless connections, said programs being implemented in a programming language for which the source code may be compiled to an intermediary code, said intermediary code being executable and verifiable, and interpreted or translated to a machine code:

said apparatus comprising:

- memory means for storing programs in the form of machine code, in such a way that they can be executed in at least one of the types of client terminals in the network.
- means for transferring a program, in the form of machine code, to at least one client terminal.

According to a preferred embodiment, said apparatus also comprises at least one translator for the intermediary code, said translator being adapted to at least one hardware and software environment used by terminals in the network.
According to this embodiment, the method also comprises the step of translating said program using said at least one translator in the host before storing them.

Programs may be downloaded to a terminal automatically or on a request from the user of the terminal.

The method is particularly useful for portable terminals in a cellular telecommunications network.

According to a preferred embodiment, the appropriate program variant to be downloaded to the terminal is determined on the basis of the type of terminal, for example, on the basis of an identity code of the terminal.

The objects are also achieved by a portable terminal adapted to wireless communication comprising a core, arranged to receive programs and/or program parts in the form of machine code, and to install the received programs during operation.

The portable terminal may be arranged to provide the user with information about the dependency tree of the available programs.

The portable terminal may also be arranged to inform the appropriate node in the network about its type and the subscription type with which it is currently operating, and other information, such as its current battery status.

According to a preferred embodiment, the portable terminal according to the invention also receives and interprets information about whether or not a program or program block should be stored in the portable terminal.

The most common language of the above mentioned kind today is Java. Other examples are LISP, SmallTalk and Erlang. The source code of a Java program is not compiled to form machine code for a certain type of processor, but to short pseudo-instructions for a virtual machine, "byte code". When a program is to be executed, the byte code is typically downloaded to the terminal on which it is to run and then translated into machine code for said terminal. In this way, a program written in an interpreted language, such as Java, can be executed in any kind of computer that has a translator or an interpreter for the language.

Java is primarily used in Internet applications, but the advantages of interpreted languages also make them feasible for use in cellular networks. Java programs are

usually smaller than corresponding machine code programs and are therefore suitable for mobile terminals, in which the memory space available is limited. An increasing number of mobile terminals support Java applications.

The use of an interpreted language, such as Java, is particularly feasible in portable terminals because their memory space is limited. Translating the program outside of the terminal also reduces the requirements on processing speed and power, which are also limited in portable terminals.

One main advantage of the present invention is that it enables a division of the responsibility for carrying out the different functions involved in changing the functionality of a device. For example, a software manufacturer can be responsible for developing the source code of the program and for compiling the source code into byte code. The manufacturer of a mobile terminal can be responsible for translating the byte code into machine code and the network operator can be responsible for the transfer of the machine code to the terminals on which it is to be run.

Interpreted languages, such as Java, have a number of advantages:
- They make it possible to load and unload portions of code dynamically to reduce the amount of space needed,
- They facilitate exchanging parts of the software since dynamic linking of programs is easier than with other program languages, and
- They make it possible to run the same byte code on different processors and with different system architectures, so that a particular function only has to be programmed once.

The last two points are useful for all portable devices, and especially for portable devices that can communicate with a wireless network of some kind, which will be able to utilize all types of services offered by the network. Also, upgrading the software by downloading new programs will be easy, so that the terminal will not

become obsolete as the functions are upgraded or new services are developed. This will also make it easier to change from one operator to the other, in which case the service package of the new operator will replace the old package. It is also perceivable to let the services provided by a mobile operator change according to, for example, the time of day, the error status of the base stations in the area, the availability of Internet-related services, etc.

Brief Description of the Drawings

Figure 1 is a simplified view of a network in which the solution according to the invention is applied.

Figure 2 is a more complete view of a network in which the solution according to the invention is applied.

Figure 3 is a flow chart of the procedure of making a new service available in the network according to the invention.

Figure 4 is a flow chart of the procedure of downloading a new service to a terminal automatically according to the invention.

Figure 5 is a flow chart of the procedure of downloading a new service to a terminal on a user's request according to the invention.

Figure 6 is an overview of how the responsibility may be divided between different participants according to the invention.

Detailed Description of Embodiments

Figure 1 is a simplified view of a network in which the solution according to the invention is applied. A portable terminal 1 is connected, via an air interface to a mobile network, through a base station 3. The base station is connected to a control unit 5, which is in turn connected to a network (not shown). The control unit 5 controls the base stations, and performs resource allocation functions and other functions such as switching functions in the mobile telecommunications network. In, or in connection to, the control unit 5, there is a program provisioning unit 9.

The control unit 5 is connected to one or more hardware vendors' hosts 7, in turn connected to hosts 8, for example belonging to software vendors. Typically, the software vendors' hosts 8 are used for producing the source code and for compiling the source code into byte code. The byte code is then transferred to the hardware vendors' hosts 7, in which the byte code is translated into machine code. For this purpose, each hardware vendor's host 7 comprises one or more translators 11 for at least the terminal types of the particular hardware vendor. In both the hardware vendors' hosts 7 and the software vendors' hosts 8 test activities will normally be carried out in ways known in the art.

In the discussion above, the division of functions into the hardware vendors' hosts 7, the software vendors' hosts 8 and the control unit 5 or program provisioning unit 9 is mainly done to clarify the different functions performed. Of course the hosts 7, 8 may also be one host in which the source code is both compiled and translated. Also, the translating function may be carried out in the program provisioning unit 9, in which case the program provisioning unit must comprise at least one translator 11.

The program provisioning unit 9 comprises a number of programs 13, 15. Each program exists in a number of variants, 13a, 13b, 13c, 15a, 15b, 15c, translated by different translators 11 to be executable in different types of terminals 1.

Alternatively, only the byte code is stored. In this case, when a program 13, 15 is to be downloaded to a portable terminal 1, this program 13, 15 is translated by the appropriate translator 11 in order to be executable on the terminal 1 and the resulting machine code is transmitted to the terminal 1. This procedure is discussed in detail in connection with Figures 4 and 5. Preferably, either the program provisioning unit 9 or the control unit 5 comprises a selection unit comprising logic for determining the appropriate variant 13a, 13b, 13c, 15a, 15b, 15c of the program 13, 15 to be

transmitted to the terminal 1. In Figure 1 such a selection unit 17 is shown in the program provisioning unit.

Figure 2 is a more extensive representation of a network in which the solution according to the invention is implemented.

The network may comprise several networks, for example, a telecommunications network 21 and a data network 23, such as the Internet. In Figure 2, the telecommunications network 21 is a cellular telecommunications network. The telecommunications network 21 comprises one or more communication units 25, such as base stations, connected to the network through control nodes 26 controlling the function of the communication units 25 and possibly other functions, such as switching functions in the network. The communication units 25 are adapted to wireless communication with terminals 27 in the network.

The data network 23 also comprises a number of terminals 29, 31, some of which 29 are used to provide programs in the form of source code or byte code. Others 31 may retrieve programs that are made available according to the invention.

The control nodes 26 also comprise the programs to be downloaded to the terminals 27 in the cellular network 21. In a preferred embodiment, the translated versions of the programs are downloaded to the control nodes 26 and stored there. The programs may also be downloaded to the control node 26 as byte code, translated in the control node 26 and stored there as machine code. The programs may also be downloaded to the control node 26 and stored as byte code. In the latter case, the byte code must be translated to machine code when the program is to be downloaded to a terminal.

As will be understood, any terminal 27, 31 connected to any of the networks 21, 23 can also download programs from any host 25, 29 in any of the networks, provided

the functions required to do so are available in the terminal 27, 29, 31 or in the network.

In this embodiment a selection unit 33 is shown in each control node 26. The selection unit 33 comprises logic for selecting the appropriate variant of a program to be transmitted to a terminal 27. If translated programs are stored, the selection unit may also comprise have access to a list of the appropriate variant of a program for each terminal type and/or subscription. If the byte code is stored and the program translated when it is to be transmitted, the selection unit may comprise or have access to a list specifying which translator should be used for transmission to a particular terminal type and/or subscription. The list may of course also state that a particular terminal type and/or subscription should not receive any variant of a particular program.

Figure 3 is a flow chart of the steps taken to make a new program available in the network.

Step S31: The source code of the service program is written, in Java or another interpreted language.

Step S32: The source code produced in step S31 is compiled to produce byte code. This step will normally include security checks, error detection, function testing, debugging, and so on.

Step S33: The byte code produced in step S32 is downloaded to the server on which it should be stored. This step will normally also include security checks.

Step S34: The service programs are translated by each of the translators available. If the service programs are not to be run on all types of terminals, of course, they only have to be translated by the relevant translators.

Step S35: The machine code resulting from step S34 are stored in a place from which it can be downloaded to portable terminals or other devices.

If the storage space of the control node 26 is limited, it may not be feasible to store several different translated versions of the program. In this case, program may be stored in the form of byte code. The translation of the byte code then takes place when it is downloaded to the terminal, that is, steps S34 and S35 are not needed. Of course, it will be possible to interpret the programs before they are downloaded to the server on which they are to be stored, that is, to change the order of steps S33 and S34.

Figure 4 is a flowchart of the steps performed when a service is to be automatically downloaded to a portable device.

Step S41: The system informs the user that it may be desirable to download a program and requests an acknowledgement. This may be, for example, because a new program, or a new version of a program has been made available, or because the user selects a function that requires a program that is not found in the user's terminal.

Step S42: The user accepts or denies the downloading of the new program version.

Step S43: If the user accepts the downloading, the appropriate program variant for the terminal type is selected and the program is downloaded. End of procedure.

Figure 5 is a flowchart of the steps performed when a service is to be downloaded to a portable terminal on request of a user for execution or storing of a program.

Step S51: The user initiates a connection between the device to which the program should be downloaded and the server on which the program is located. This may be done in any way known in the art, through an air interface or a wired connection.

Step S52: The type of terminal is determined. This is preferably done automatically by the system, but if necessary, the user may specify the type of terminal.

Step S53: If appropriate, the user selects the program or programs to be downloaded.

Step S54: The appropriate variant and version of the program or programs selected in step S53 is identified and downloaded. If the programs are stored as byte code in the server, they must also be translated by the appropriate translator when being downloaded. End of procedure.

The programs may be temporarily stored in the terminal and discarded once they have been executed, or the programs may be stored in the terminal.

In steps S43 and S54 the selection of the appropriate program variant may be carried out in the communication unit, or this information may be retrieved from the vendor of the terminal or the software vendor.

The programs, and the appropriate variant of each program, that may be downloaded to a particular terminal may be determined in dependence of the type of terminal and the type of subscription.

Information about the type of subscription is found on a chip in the telephone, and is always communicated to the base station when a connection is to be set up. This chip, for example, the SIM card in GSM, is normally removable and may inserted in a number of different terminals. Thus, the type of terminal cannot be determined from the information on the chip. Different terminal types may, however, be able to run different programs, and require different variants of programs, even if the subscription type is the same.

The terminal type may be determined both automatically and by manual identification. In the latter case, for example, each terminal type may be assigned an identification number, which is entered every time a program is to be downloaded.

In a preferred embodiment, the terminal type is determined using an identity code of the terminal, which is a unique number identifying a wireless terminal. In GSM, for example the International Mobile Station Equipment Identity (IMEI) might be used. This number (IMEI) is stored in every mobile terminal and should in this case be transmitted by the mobile terminal to the network to indicate the variant of a program that whould be received. It would also be possible to define a number of terminal types and enable every terminal to inform the system about its terminal type. In order for this to work, information about the requirements of each series of terminals must be supplied from the manufacturers of the terminals and stored in each host, or retrieved from the manufacturer each time it is needed. This information may have the form of a specification of the translator that should be used for each terminal type. The translator may be provided by the terminal manufacturer, in which case the appropriate translator only has to be identified. If the terminal manufacturer does not provide the translator, the requirements on the translator must be specified in detail.

The programs to be downloaded may also be determined in dependence of other factors, such as the cell identity, the operator, the date or the time of the day.

Automatic downloading of programs may be used, for example, in the following situations:

- When a program package has been updated it may be downloaded to each terminal the first time it connects to the network after the update.
- When the user tries to activate a service, the program associated with this service may be downloaded without the user being explicitly informed first.
- Certain programs may be automatically downloaded in dependence of the location, the time of the day, the operator, etc.

In the first case listed above, it should be possible for the user to stop the downloading of software, if he/she does not want to receive new software. This may happen,

for example because of lack of time or because the battery is low. The downloading may also be stopped automatically, for example, if the battery is low.

The situation when a subscriber is connected to another network than the one in which he/she is a subscriber, for example, when abroad, may be handled in different ways. In this case, the subscription information transmitted will let the system know that the subscriber does not belong in the current network. Even if automatic downloading may not be feasible, it may be desirable to allow the downloading of software on request

The transmission of programs from the host to the terminal may be made according to existing protocols for data transmission in the system concerned. For example, in GSM, the protocol for packet communication, Global Packet Radio Service (GPRS) may be used. In Advanced Mobile Phone Service (AMPS) systems, The Cellular Digital Packet Data (CDPD) protocol may be used. The latter protocol also comprises functions for suspending data communication when a voice call is attempted.

The transfer of the program to the mobile terminal may be carried out in any available channel. For transferring a program from the communication unit to one mobile terminal at a given time, a traffic channel may be used. Of course, data may be transmitted according to circuit switched protocols or according to packet switched protocols, such as GPRS in a GSM system. Also, high-speed connections involving more than one traffic channel may be used, for example High-Speed Circuit-Switched Data (HSCSD) in GSM. This will make the transfer of the program faster but will probably only be feasible at times when the traffic load in the network is low.

Broadcasting may be used when the same program is to be transferred to several mobile terminals at the same time. In this case, control information must be added to let the mobile terminals know which terminal types are to receive the program, and

each terminals must comprise logic to be able to determine whether or not the program is intended for this particular terminal.

In some situations it will be feasible to store parts of the program code temporarily or permanently, especially parts that are to be run several times, to avoid having to download them every time they are to run. For example, subroutines of a program that are to be run several times during the execution of the program may be stored temporarily in the portable terminal for as long as the program is running. Programs that perform important functions may be stored permanently or semipermanently.

The program parts to be stored in this way may be selected by the subscriber or by a node in the network, but preferably by both in cooperation. Several ways of achieving this are known in the art. For example, each program or program part may be marked, to signify if it should be stored in the mobile terminal from a system point of view, for example, to save bandwidth. To determine whether or not a program or program part should be stored in the mobile terminal, however, knowledge of the state of the terminal is also needed. Therefore, it is usually not feasible to store these program parts automatically, without giving the subscriber a chance to interrupt.

If the subscriber is to delete a program or a program part, he/she must be given information about the dependency tree, that is, what other programs or programs part use the program or program part to be deleted, in order to decide if the program or program part should really be deleted.

The mobile terminal should therefore contain functions enabling the subscriber to determine whether or not a specific program or program part should be stored, and for how long, and providing the information needed by the subscriber to make a decision.

Figure 6 illustrates the division of the responsibilities for providing the programs according to the invention.

A service provider 101 provides the appropriate source code, or byte code. To do this, the service provider 101 must know the desired functionality of the service to be provided. Detailed information about the software and hardware environment provided in the terminals is not needed, but basic knowledge about, for example, the capacity of a terminal may be useful.

A terminal manufacturer 103 provides the terminals and, usually, a translator for the programming language, adapted for the terminals he provides. The terminal manufacturer 103 must have knowledge about the programming language used. There may of course be several service providers and terminal manufacturers.

A network operator 105 is responsible for a mobile network 107 and for providing the services to subscribers 109 in the network 107.

The responsibilities may be divided between the service provider 101, the terminal provider 103 and the network operator 105 in a number of different ways. For example, the service provider 101 may provide byte code to the terminal manufacturer 103, as indicated by the arrow 111. The terminal manufacturer 103 then translates the byte code to machine code using the appropriate translators for the terminals he provides, and provides this machine code to the network operator 105 through a transport network 113, such as the Internet, as indicated by an arrow 115. Of course, the machine code could be provided in any way known in the art, for example, by means of a direct connection or on a floppy disk, or CD-ROM.

Another way of dividing the responsibilities would be for the service provider 101 to provide the byte code directly to the network operator 105, through the transport network 113, as indicated by an arrow 117. The terminal manufacturer 103 could

provide the translator to the network operator 105, and the network operator 105 could translate the byte code into machine code.

According to both these methods the machine code could be stored in a memory 119 and provided to the subscriber 109 through the cellular network 107 when desired, as indicated by an arrow 121. Of course, the service provider 101 could deliver the source code instead of the byte code, but it is normally not desirable to deliver source code.

Claims

1. A method of changing the functionality of a mobile terminal connected by a wireless connection in a network,
said functionality being implemented as programs in a programming language for which the source code may be compiled to an intermediary code, said intermediary code being executable and verifiable,
characterized by the steps of
   - translating said intermediary code into machine code using at least one translator in dependence of the hardware and software environment of at least one terminal type;
   - determining the appropriate variant of the machine code to be downloaded to the terminal;
   - downloading the appropriate variant of the machine code to said terminal.

2. A method according to claim 1, **characterized** by
   - translating said program into machine code using said at least one translator;
   - storing the machine code of said programs in a host in the network
   - transferring said machine code to at least one of said terminals when desired.

3. A method according to claim 1 or 2, wherein the downloading of programs is initiated by a unit in the network.

4. A method according to claim 1 or 2, wherein programs are downloaded to a terminal on a request from the user of the terminal.

5. A method according to any one of the preceding claims, wherein at least some of said terminals are portable terminals for use in a cellular telecommunications network.

6. A method according to any one of the preceding claims, **characterized** by the step of determining whether or not a program block should be stored temporarily or permanently in the terminal and, in this case, storing the program accordingly.

7. A method according to any one of the preceding claims, **characterized** by the step of determining the appropriate variant of the machine code to be downloaded to the terminal on the basis of an identity code of the terminal;

8. A method according to any one of the preceding claims, characterized by the step of determining the appropriate variant of the machine code to be downloaded to the terminal on the basis of a code identifying the type of subscription.

9. A method according to any one of the preceding claims, **characterized** by the steps of
   - for each program, determining the terminal types that can use the program;
   - translating the program using the translators relevant for said terminal types.

10. An apparatus for providing programs in a network comprising hosts and several different types of terminals, at least some of said terminals being connected to other parts of the network by wireless connections, said programs being implemented in a programming language for which the source code may be compiled to an intermediary code, said intermediary code being executable and verifiable, and interpreted or translated to a machine code:
    said apparatus being **characterized** in that it comprises:
    - memory means for storing programs in the form of machine code, executable in at least one of the types of terminals in the network;
    - determining which machine code should be downloaded to a particular terminal on the basis of the terminal type;
    - means for transferring a program, in the form of machine code, to at least one terminal.

11. An apparatus according to claim 10, **characterized** in that it comprises translation means for translating intermediary code into machine code in such a way that it can be executed in at least one type of portable terminal in a cellular network.

12. An apparatus according to claim 10 or 11, **characterized** in that at least one type of terminal is a portable terminal connected in a cellular network.

13. An apparatus according to any one of the claims 10-12, **characterized** in that it comprises means for determining the terminal types that can use each program.

14. An apparatus according to claim 13, **characterized** in that it comprises means for determining the machine code that should be downloaded to a particular terminal on the basis of a serial number of the terminal.

15. An apparatus according to any one of the claims 10-14, **characterized** in that it comprises means for determining the program that should be downloaded to a particular terminal on the basis of the type of subscription applying to the terminal.

16. A telecommunications network comprising hosts and several different types of terminals, at least some of said terminals being connected to other parts of the network by wireless connections, **characterized** in that it comprises:

- at least one translating means for a programming language for which the source code may be compiled to an intermediary code, said translating means being used to translate said intermediary code into machine code executable on at least one of said terminals;
- memory means for storing the machine code;

- means for determining what programs or program versions should be downloaded to a particular terminal
- means for transferring a program, in the form of machine code, to at least one terminal.

17. A telecommunications network according to claim 16, **characterized** in that it comprises means for determining what programs or program versions should be downloaded to a particular terminal on the basis of the terminal type.

18. A telecommunications network according to claim 16 or 17, **characterized** in that it comprises means for determining what programs or program versions should be downloaded to a particular terminal on the basis of the type of subscription.

19. A portable terminal adapted to wireless communication comprising a core, adapted to receive programs and/or program parts in the form of machine code, **characterized** in that said core is arranged to install the received programs and/or program parts during operation, and that it is arranged to inform the appropriate node in the network about its type and/or the subscription type with which it is currently operating.

20. A portable terminal according to claim 19, **characterized** in that it is arranged to provide the user with information about the dependency tree of the available programs.

21. A portable terminal according to claim 19 or 20, **characterized** in that it is arranged to inform the appropriate node in the network about its type and the subscription type with which it is currently operating.

22. A portable terminal according to any one of the claims 19-21, **characterized** in that it is arranged to inform the appropriate node in the network about its current battery status.

23. A portable terminal according to any one of the claims 19-22, **characterized** in that it is arranged to receive and interpret information about whether or not a program or program block should be stored in the portable terminal.

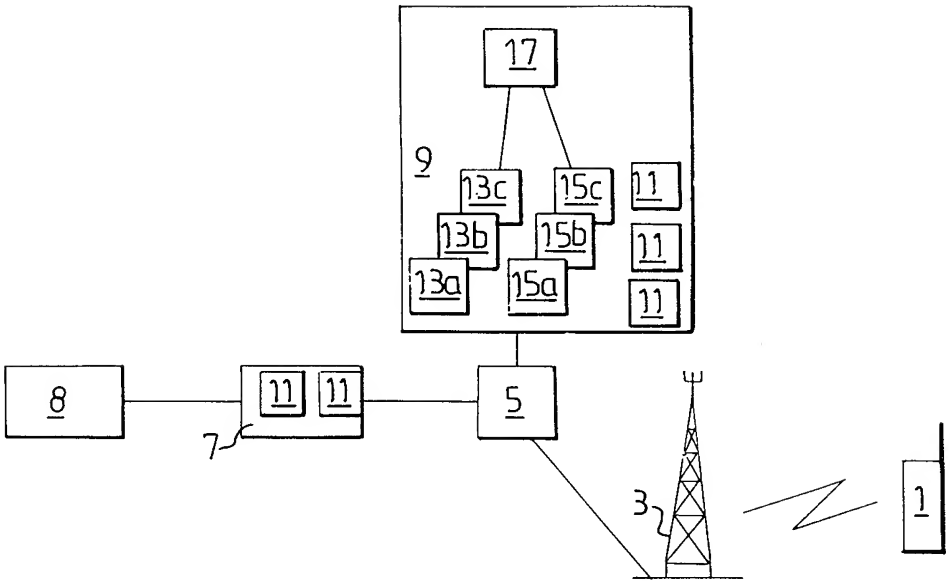

17
9
13c
15c
11
13b
15b
11
13a
15a
11
8
11
11
5
7
3
1

FIG. 1

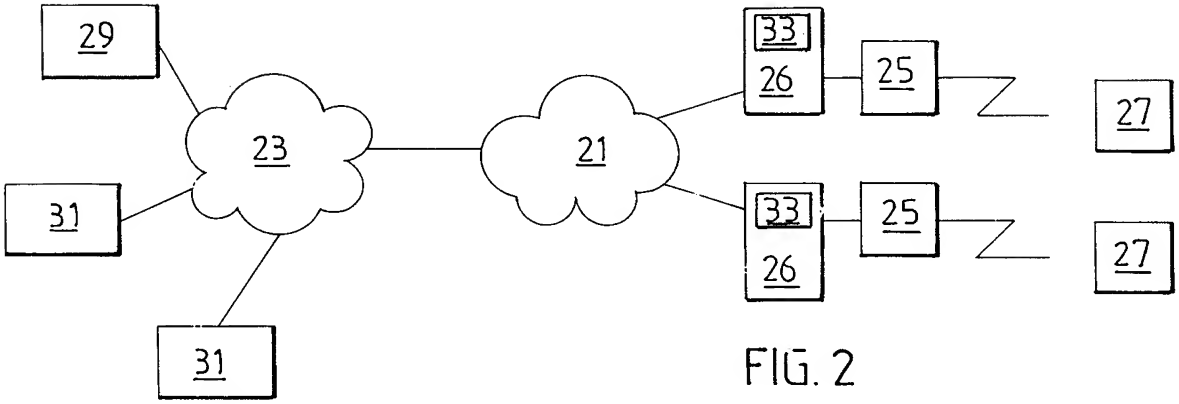

29
23
31
31
21
33
26
25
27
33
26
25
27

FIG. 2

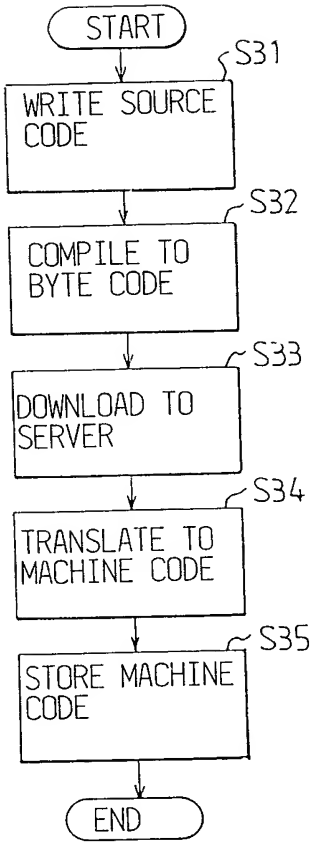


FIG.3

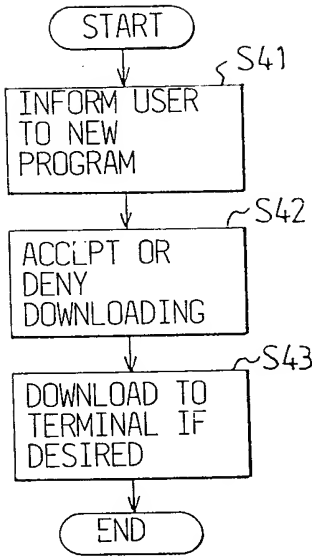


FIG.4

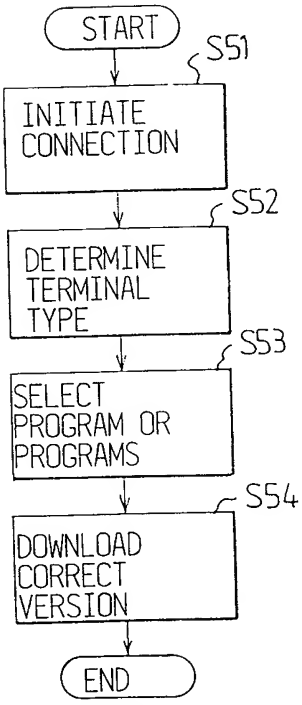


FIG.5

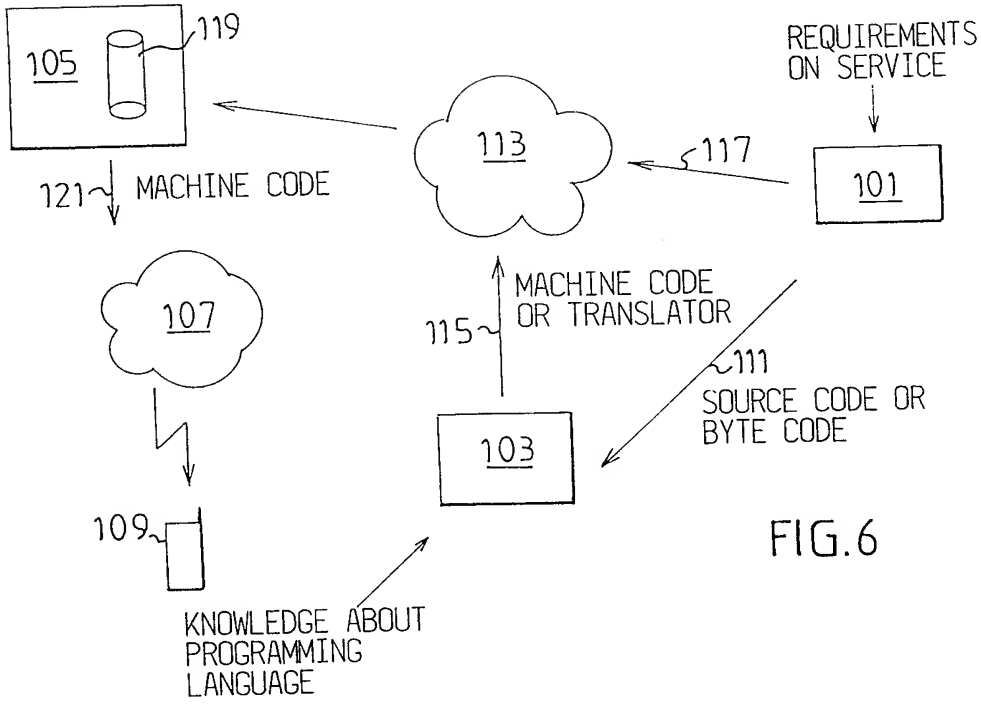


FIG.6